Icebridge portal

What We're Trying To Do

The IceBridge Portal shows a polar stereographic projection of either the northern or southern hemisphere, depending on the user's interest. For the selected hemisphere, we show a list of collections that are (a) IceBridge datasets, and (b) have a bounding box that intersects with the hemisphere they are viewing (e.g., for the northern hemisphere, a bounding box of [-180, 0, 180, 90]). For each collection, we show two counts: (i) the number of granules that match the user's current temporal and spatial filters, and (ii) the total number of granules in the hemisphere for that collection. So it looks something like this if the user hasn't set any temporal or spatial filters:

Dataset	Granules in constraint	Granules in hemisphere
IAKST1B Version 001	123	123
IDCSI4 Version 001	46	46
IDHDT4 Version 001	204	204

When they change their temporal or spatial filters, we'd like to update the first count--the count showing the number of granules for those filters, e.g.:

Dataset	Granules in constraint	Granules in hemisphere
IAKST1B Version 001	13	123
IDCSI4 Version 001	0	46
IDHDT4 Version 001	93	204

Along with this list view, there is a map which displays granules that they've selected to view. For now, I'm not interested in the granule queries we're doing against CMR.

What We're Doing

Currently we're doing something bad and inefficient in IceBridge Portal. This was an initial attempt to get feedback to the user more quickly so they could see a more responsive interface. So basically I'm apologizing in advance for what I'm about to say .

We do three types of queries to populate the list with counts shown above:

- To get the list of collections, we do the following CMR query **once** when the user first arrives at the portal:

```
https://cmr.earthdata.nasa.gov/search/collections.json?keyword=icebridge&page_size=100&temporal=2009-01-01T07:00:00.000Z,2016-01-28T16:19:34.258Z&bounding_box=-180,0,180,90
```

- To get the **total** number of granules for each collection in the hemisphere the user is currently viewing, we issue a **separate query** to CMR like the following:

```
https://cmr.earthdata.nasa.gov/search/collections.json?keyword=icebridge&page_size=100&bounding_box=-180,0,180,90&include_granule_counts=true&concept_id=C1000000341-NSIDC_ECS
```

This gets the number of granules for the entire northern hemisphere for one specific collection. So here's the bad part: we issue this query **for**

each collection returned from query #1 above (!). For IceBridge, this is 50-60 queries.

Whenever a user changes temporal or spatial filters, we again issue query #2, but with their temporal and spatial filters, e.g.:

```
https://cmr.earthdata.nasa.gov/search/collections.json?keyword=icebridge&p
age_size=100&temporal=2009-01-01T07:00:00.000Z,2016-02-01T21:15:36.639Z&po
lygon=-54.28856753366417,70.20590793535574,-53.76512809420709,69.058171675
34093,-51.035282396612224,69.18466321053025,-51.39895662736889,70.34040480
5645,-54.28856753366417,70.20590793535574&include_granule_counts=true&conc
ept_id=C1000000180-NSIDC_ECS
```

Again, we know this is bad, but we do a separate query for each collection in their list (~50-60 collections / queries). So doing queries 2 and 3, even though we issue lots of queries, the users see results coming back immediately, rather than waiting 5 seconds to get all the results.

What We'd Like To Do

- Issue **one** CMR query to get a list of IceBridge collections in the hemisphere, along with the granule counts. E.g.:

```
https://cmr.earthdata.nasa.gov/search/collections.json?keyword=icebridge&p
age_size=100&temporal=2009-01-01T07:00:00.000Z,2016-01-28T16:19:34.258Z&bo
unding_box=-180,0,180,90&include_granule_counts=true
```

(just query #1 with granule counts) We did start with this, but the query was taking 10-20s at the time, IIRC. That's why we switched to the set of queries shown above. It seems now that this query runs quite a bit faster than it did. So it's quite possible that we could switch back to it (see below).

When the user changes their filters, issue **one** CMR query to get a list of IceBridge collections with a specific temporal and spatial filter, along with matching granule counts. E.g.:

```
https://cmr.earthdata.nasa.gov/search/collections.json?keyword=icebridge&p
age_size=100&temporal=2009-01-01T07:00:00.000Z,2016-02-01T21:15:36.639Z&po
lygon=-54.28856753366417,70.20590793535574,-53.76512809420709,69.058171675
34093,-51.035282396612224,69.18466321053025,-51.39895662736889,70.34040480
5645,-54.28856753366417,70.20590793535574&include_granule_counts=true
```

Questions

1. Are there more efficient ways of issuing these queries to CMR (e.g., other parameters, options, etc)?
2. Are there optimizations that you can do in CMR that would improve the performance of these queries?
3. Are there other ways of slicing and dicing this problem that you can think of?